

2. Redes Neurais Artificiais

Prof. Renato Tinós

Depto. de Computação e Matemática (FFCLRP/USP)

2.6. Deep Learning

2.6.1. O Problema da Extração de Características

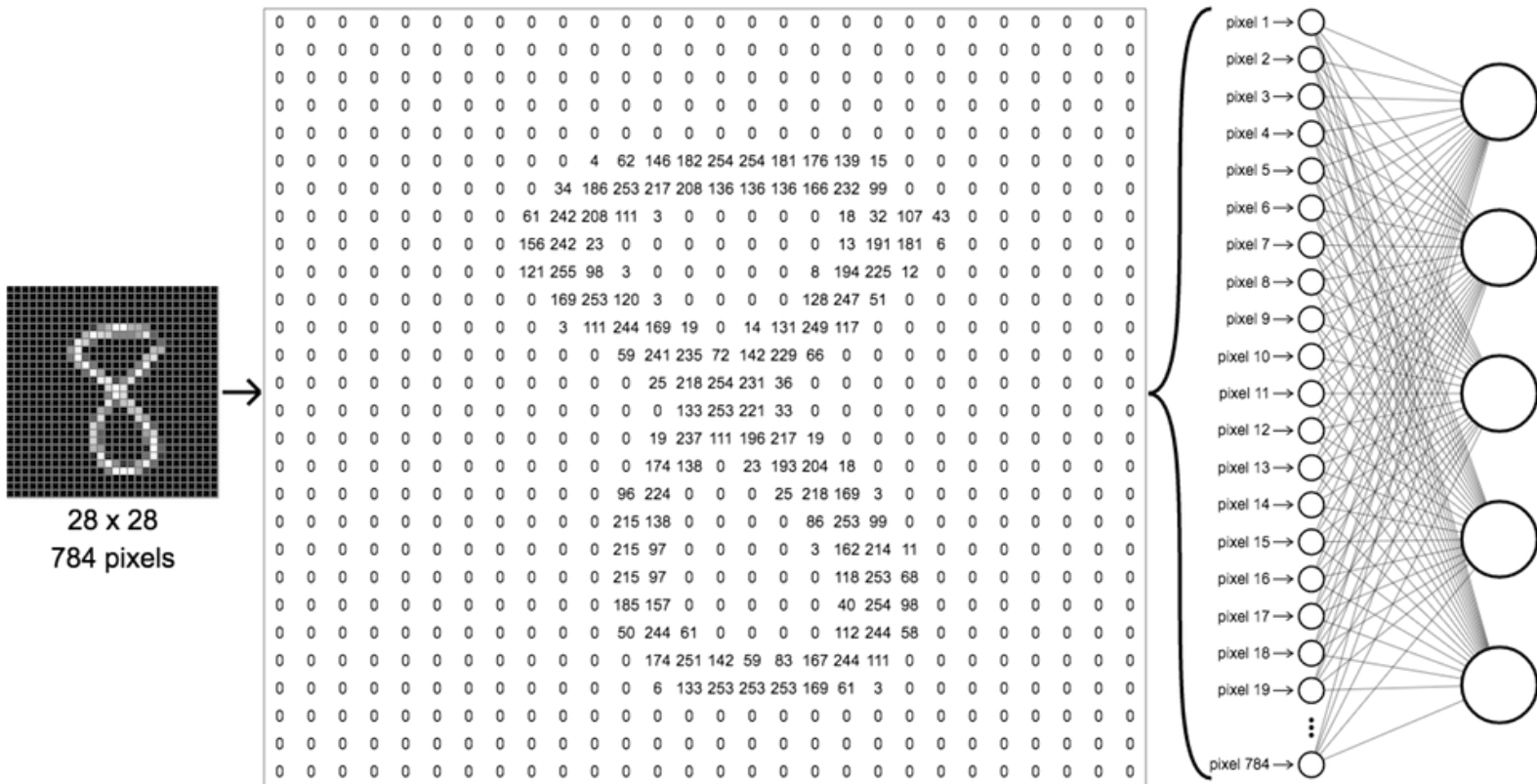
2.6.2. Vanishing Gradient Problem

2.6.3. Redes Neurais Convolucionais

2.6.1. O Problema da Extração de Características

- **Redes Neurais Convencionais**
 - Têm dificuldades para processar dados naturais em sua forma bruta
 - » Exemplos
 - ❑ Imagens
 - ❑ Texto

2.6.1. O Problema da Extração de Características



Fonte: https://ml4a.github.io/ml4a/neural_networks/

2.6.1. O Problema da Extração de Características

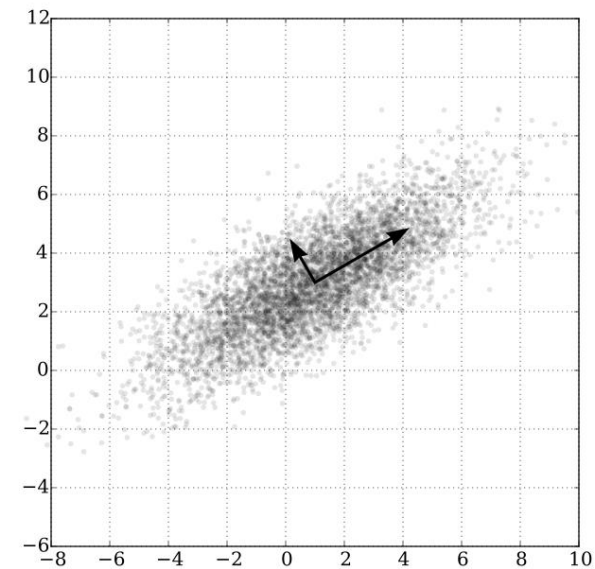
- **Redes Neurais Convencionais**

- Solução geralmente adotada: extração de características

» Exemplos em imagens:

- ☐ PCA

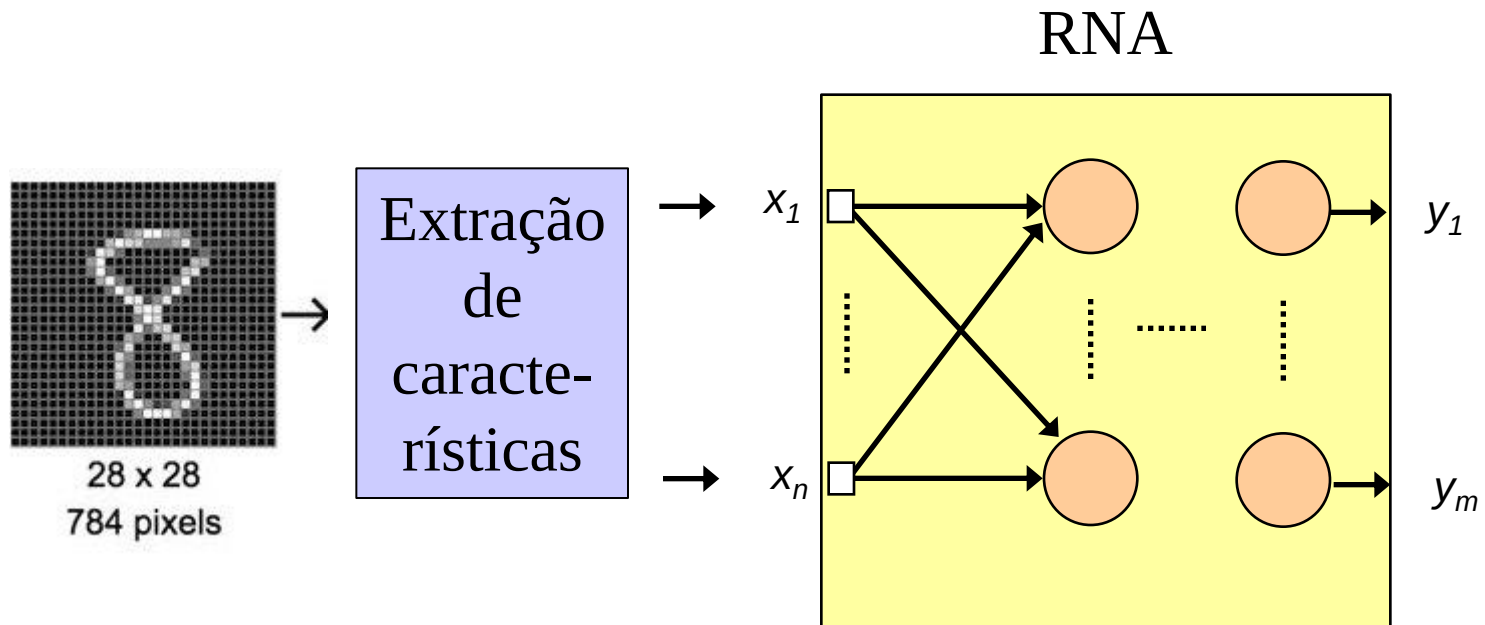
- ☐ Atributos de forma e textura



Fonte:

https://en.wikipedia.org/wiki/Principal_component_analysis#/media/File:GaussianScatterPCA.svg

2.6.1. O Problema da Extração de Características



2.6.1. O Problema da Extração de Características

- **Redes Neurais Convencionais**

- Solução geralmente adotada: extração de características

» Problemas:

- ☐ Nem sempre as características relevantes para a classificação são extraídas
- ☐ É dependente do contexto
- ☐ Muitas vezes requer um especialista para selecionar as características para determinada área

2.6.2. *Vanishing Gradient Problem*

- **Redes Neurais com muitas camadas**
 - Transformações sucessivas da informação de entrada
 - » Transformam a representação em um nível, começando da camada de entrada, em uma representação em um nível maior e ligeiramente mais abstrato

2.6.2. *Vanishing Gradient Problem*

- **Redes Neurais com muitas camadas**
 - **Problema: perda da informação do gradiente**
 - » Utilizando o Backpropagation, a informação do gradiente é propagada para trás
 - » Começando da última camada, os gradientes locais são propagados e combinados nos neurônios
 - » Quanto mais se distancia da camada de saída, mais a informação do gradiente diminui

2.6.2. *Vanishing Gradient Problem*

- **Redes Neurais com muitas camadas**
 - **Problema: perda da informação do gradiente**
 - » Este, aliado ao **crescimento do número de parâmetros ajustáveis** (pesos), são os motivos principais de redes neurais convencionais utilizarem poucas camadas
 - » O problema é ainda pior quando são utilizadas funções de ativação tradicionais como sigmoideal e tangente hiperbólica
 - ❑ O uso da regra da cadeia tem o efeito de multiplicar vários números pequenos produzidos pelas funções de ativação, fazendo com que o gradiente diminua exponencialmente

2.6.3. Redes Neurais Convolucionais

- ***Deep Learning* (Aprendizado Profundo)**
 - Métodos de aprendizagem de representação
 - » São métodos que permitem que uma máquina, ao ser alimentada com dados brutos, descubra automaticamente as melhores representações para detecção ou classificação
 - **Métodos de aprendizagem profunda são:**
 - » métodos de aprendizagem de representação com múltiplos níveis de representação
 - » São compostos por módulos simples não-lineares, que transformam a representação em um nível ligeiramente mais abstrato

2.6.3. Redes Neurais Convolucionais

- ***Deep Learning* (Aprendizado Profundo)**
 - Com a composição de tais transformações simples, funções muito complexas podem ser estimadas
 - » Para tarefas de classificação, camadas superiores de representação amplificam os aspectos das entradas que são importantes para a discriminação e suprimem variações que são irrelevantes

2.6.3. Redes Neurais Convolucionais

- ***Deep Learning* (Aprendizado Profundo)**
 - Exemplo: classificação de imagens
 - » Uma imagem vem na forma de uma matriz de valores de pixel
 - » As características descobertas na primeira camada tipicamente indicam a presença ou ausência de bordas em orientações e locais específicos da imagem
 - » A segunda camada normalmente detecta pequenos padrões particulares de bordas, independentemente de pequenas variações nas posições e orientações destas
 - » A terceira camada pode combinar pequenos padrões que resultam em padrões maiores que correspondem a partes de objetos familiares
 - » Camadas subsequentes detectariam objetos por meio da combinação de padrões obtidos anteriormente

2.6.3. Redes Neurais Convolucionais

- ***Deep Learning* (Aprendizado Profundo)**
 - O aspecto chave da aprendizagem profunda é que as características interessantes para a classificação e as transformações de representação não são diretamente projetadas por especialistas
 - » Elas são aprendidas a partir da inferência utilizando dados brutos
 - » Criando-se assim um procedimento de aprendizado de propósito geral

2.6.3. Redes Neurais Convolucionais

Samoyed (16); Papillon (5.7); Pomeranian (2.7); Arctic fox (1.0); Eskimo dog (0.6); white wolf (0.4); Siberian husky (0.4)

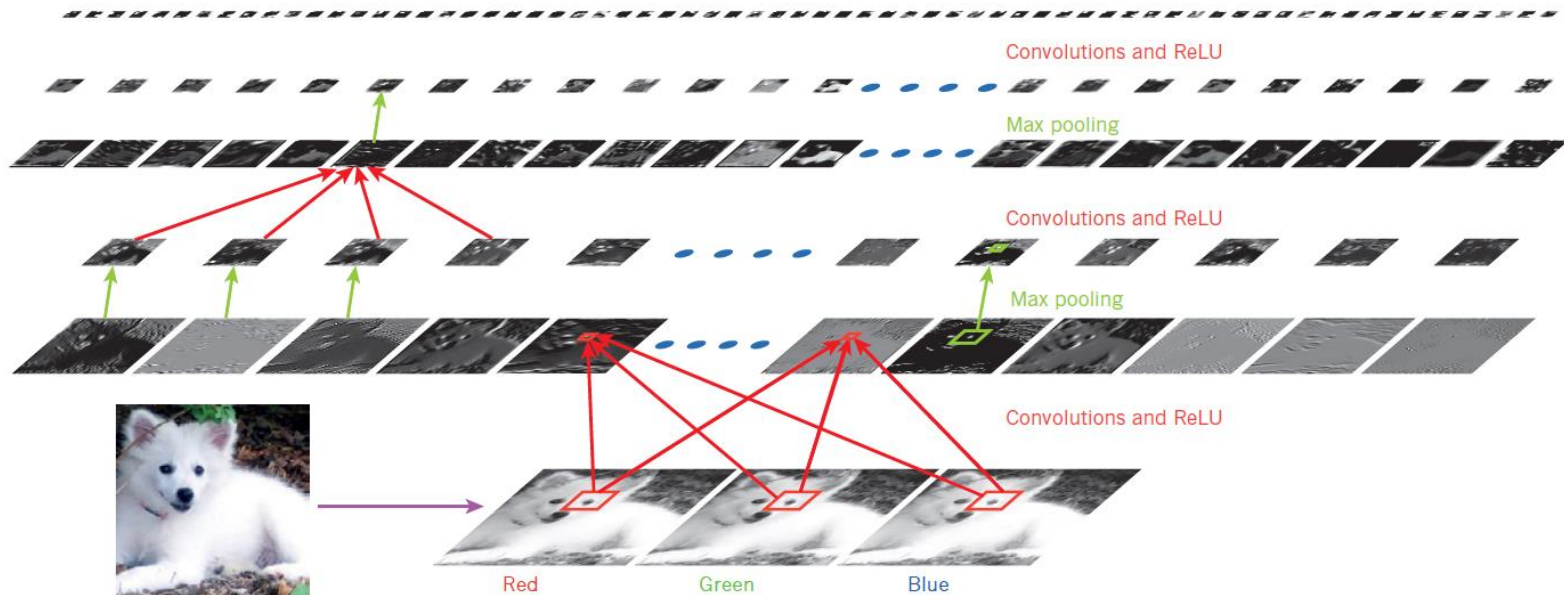
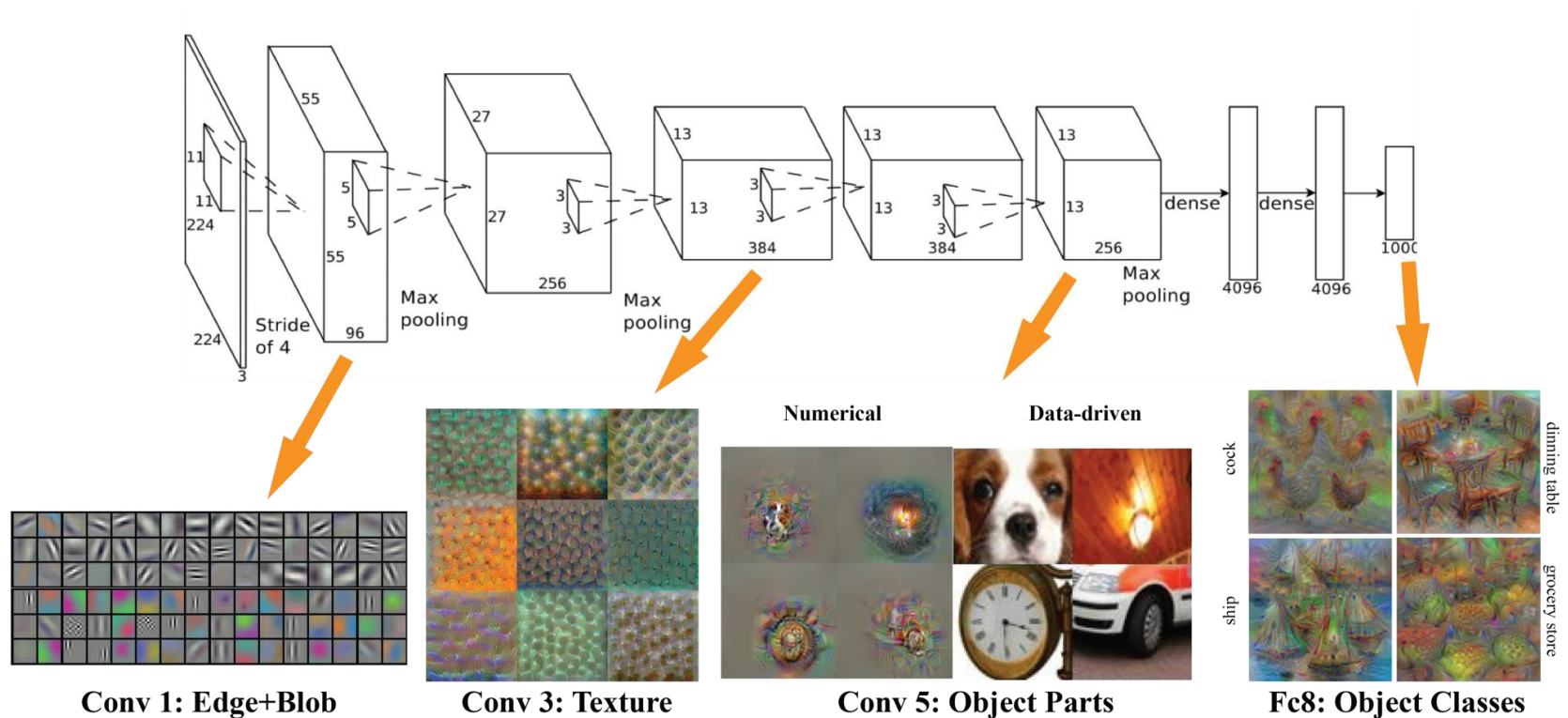


Figure 2 | Inside a convolutional network. The outputs (not the filters) of each layer (horizontally) of a typical convolutional network architecture applied to the image of a Samoyed dog (bottom left; and RGB (red, green, blue) inputs, bottom right). Each rectangular image is a feature map

corresponding to the output for one of the learned features, detected at each of the image positions. Information flows bottom up, with lower-level features acting as oriented edge detectors, and a score is computed for each image class in output. ReLU, rectified linear unit.

Fonte: LeCun, Y., Bengio, Y., & Hinton, G. (2015). "Deep learning", Nature, 521(7553), 436-444.

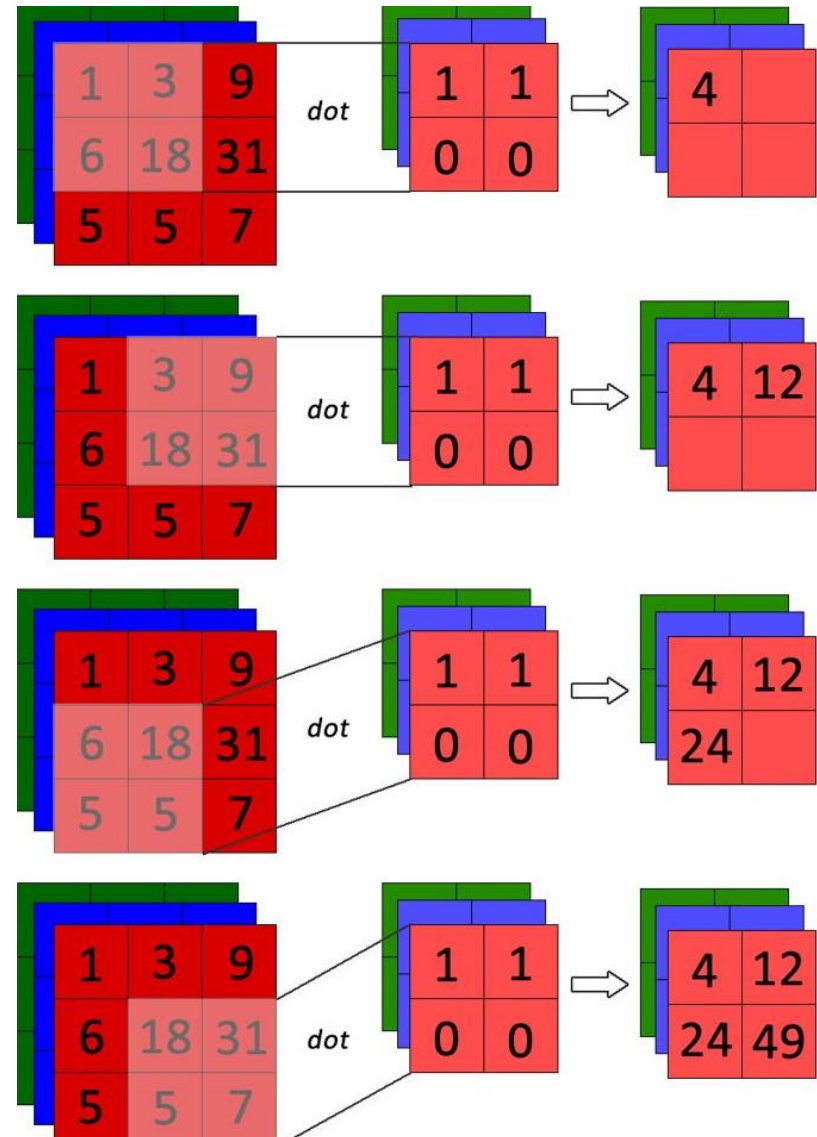
2.6.3. Redes Neurais Convolucionais



Fonte: <https://hackernoon.com/learning-ai-if-you-suck-at-math-p5-deep-learning-and-convolutional-neural-nets-in-plain-english-cda79679bbe3>

2.6.3. Redes Neurais Convolucionais

- **Camadas convolucionais**
 - Utilizam pequenos campos receptivos
 - » Diminuindo assim a complexidade



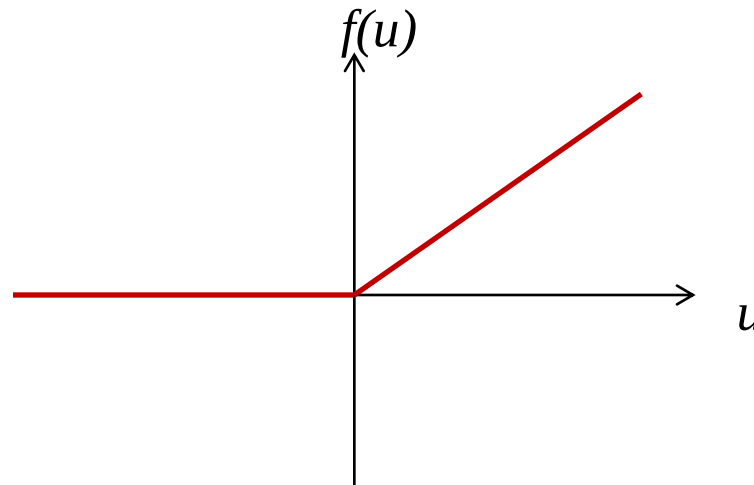
2.6.3. Redes Neurais Convolucionais

- **Camadas ReLU**

- ReLU é a abreviação para *Rectified Linear Units* (unidade linear retificada)

» A função de ativação linear retificada é dada por

$$f(u) = \max(0, u) \quad (2.6.1)$$

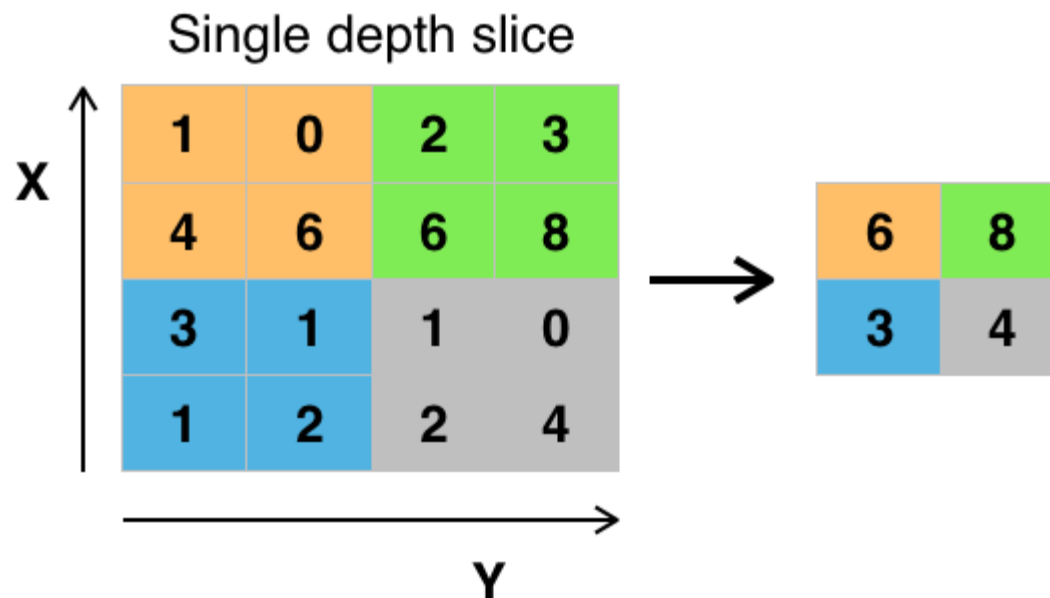


2.6.3. Redes Neurais Convolucionais

- **Camadas de Pooling**

- **Max pooling**

» Particiona a imagem de entrada em um conjunto de retângulos não-sobrepostos e, para cada sub-região, retorna o máximo



Fonte: https://en.wikipedia.org/wiki/Convolutional_neural_network#/media/File:Max_pooling.png

2.6.3. Redes Neurais Convolucionais

- **Treinamento**

- Inicialmente, foram utilizadas técnicas não-supervisionadas para as primeiras camadas
- Atualmente, utiliza-se apenas Backpropagation
 - » Quando o conjunto de treinamento tem muitos dados, diminui-se o problema dos ótimos locais
 - » Em geral, em vez de apresentar exemplos um a um, vários exemplos são apresentados para a atualização dos pesos

□ **Batch size**: número de exemplos apresentados de cada vez

2.6.3. Redes Neurais Convolucionais

- **Técnicas para minimizar o overfitting geralmente são adotadas**
 - **Aumento da base de dados**
 - » No caso de imagens, é comum aumentar artificialmente a base considerando-se imagens alteradas, por exemplo, por ampliação, redução, rotação e deslocamento
 - **Regularização de pesos**
 - » Visa reduzir problemas de mal-condicionamento
 - **Dropout**
 - » Remoção de neurônios da rede
- **Existem vários parâmetros para definir o treinamento e a operação destas redes**

2.6.3. Redes Neurais Convolucionais

ImageNet Challenge

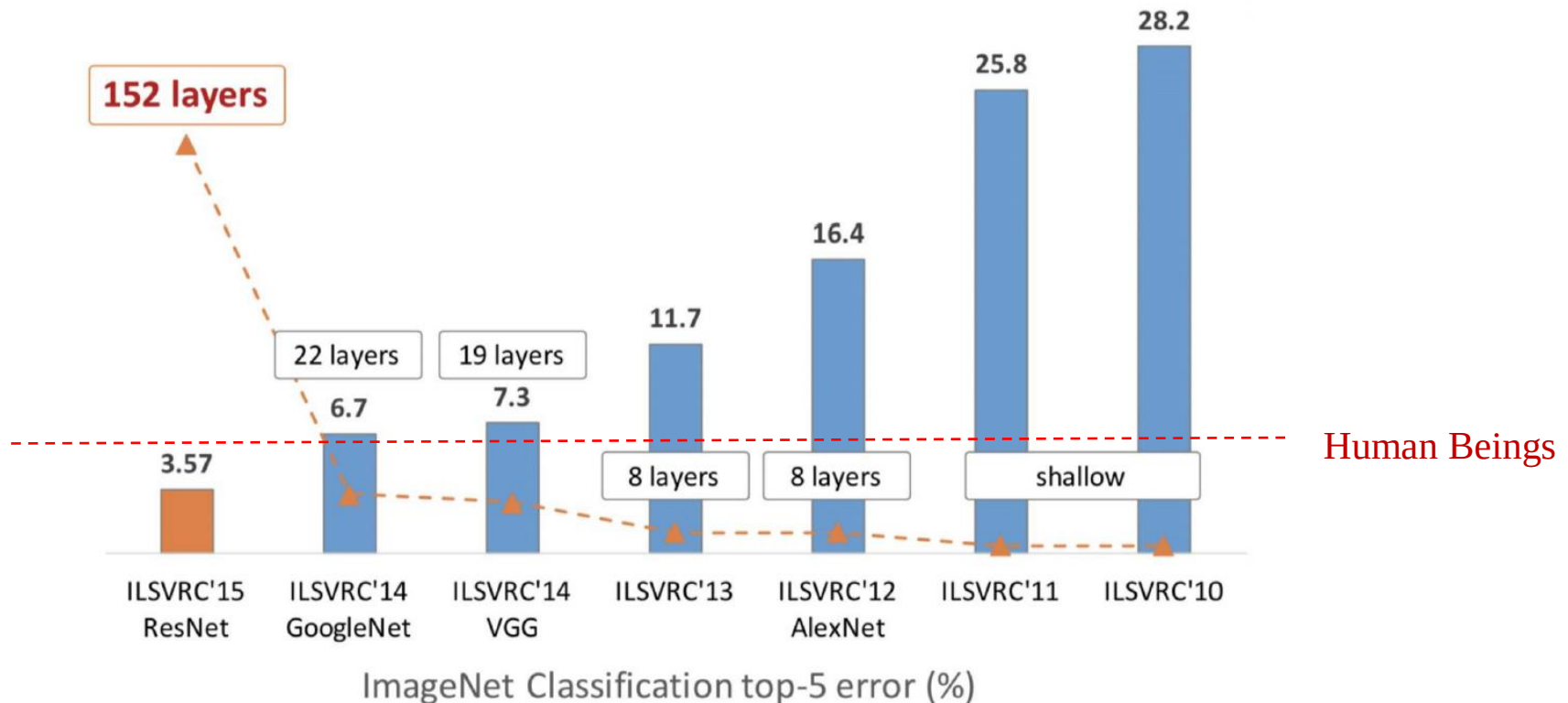
IMAGENET

- 1,000 object classes (categories).
- Images:
 - 1.2 M train
 - 100k test.



Fonte: <https://www.slideshare.net/xavigiro/deep-learning-for-computer-vision-imagenet-challenge-upc-2016>

2.6.3. Redes Neurais Convolucionais

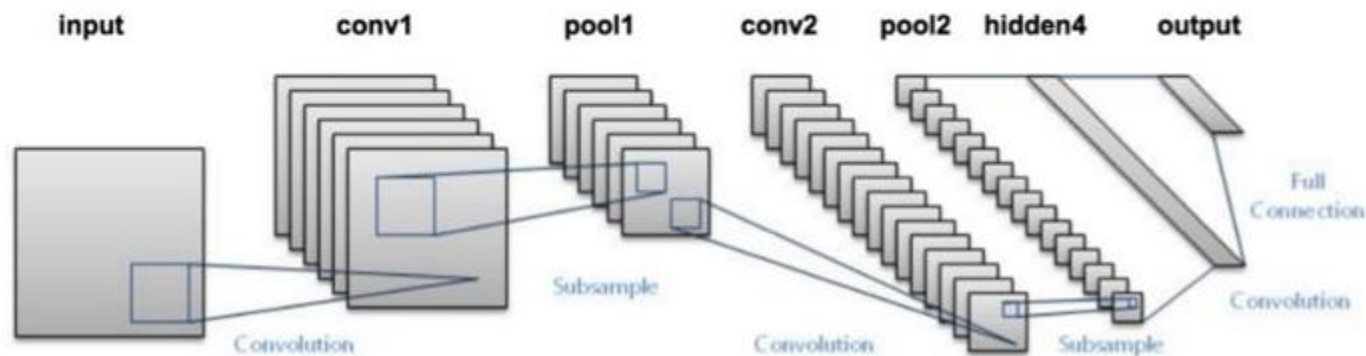


Source: He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition.

2.6.3. Redes Neurais Convolucionais

LeNet-5 (1998)

LeNet-5, a pioneering 7-level convolutional network by LeCun et al in 1998, that classifies digits, was applied by several banks to recognise hand-written numbers on checks (cheques) digitized in 32x32 pixel greyscale input images. The ability to process higher resolution images requires larger and more convolutional layers, so this technique is constrained by the availability of computing resources.

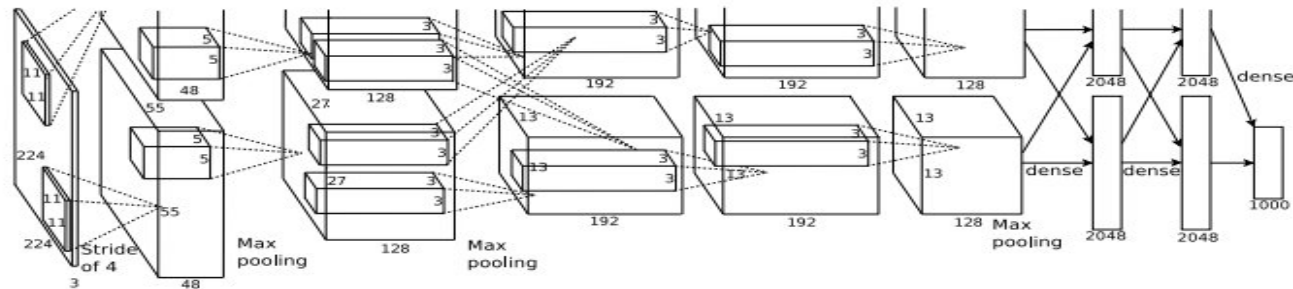


Fonte: <https://medium.com/@sidereal/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5>

2.6.3. Redes Neurais Convolucionais

AlexNet (2012)

In 2012, AlexNet significantly outperformed all the prior competitors and won the challenge by reducing the top-5 error from 26% to 15.3%. The second place top-5 error rate, which was not a CNN variation, was around 26.2%.

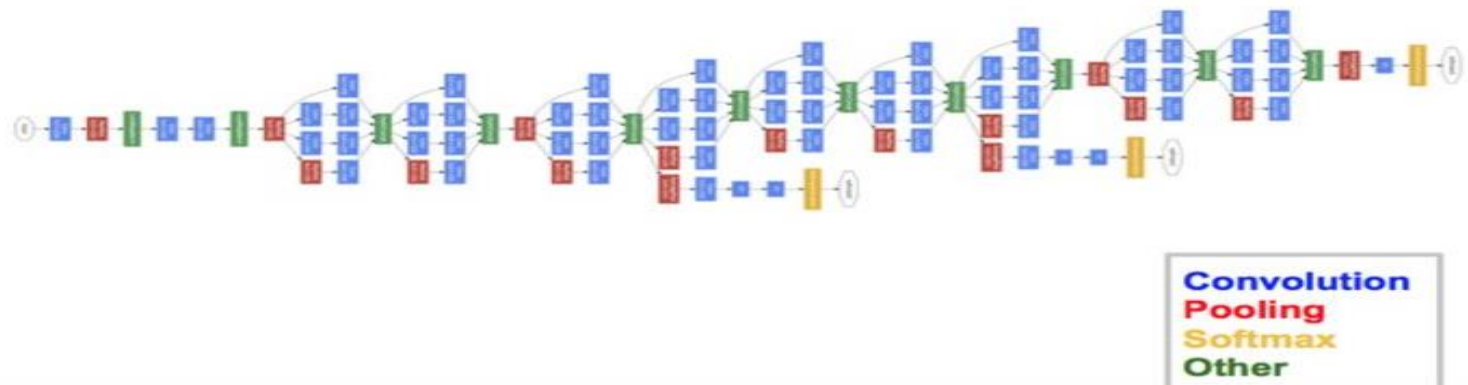


The network had a very similar architecture as LeNet by Yann LeCun et al but was deeper, with more filters per layer, and with stacked convolutional layers. It consisted 11x11, 5x5, 3x3, convolutions, max pooling, dropout, data augmentation, ReLU activations, SGD with momentum. It attached ReLU activations after every convolutional and fully-connected layer. AlexNet was trained for 6 days simultaneously on two Nvidia Geforce GTX 580 GPUs which is the reason for why their network is split into two pipelines. AlexNet was designed by the SuperVision group, consisting of Alex Krizhevsky, Geoffrey Hinton, and Ilva Sutskever.

2.6.3. Redes Neurais Convolucionais

GoogleNet/Inception(2014)

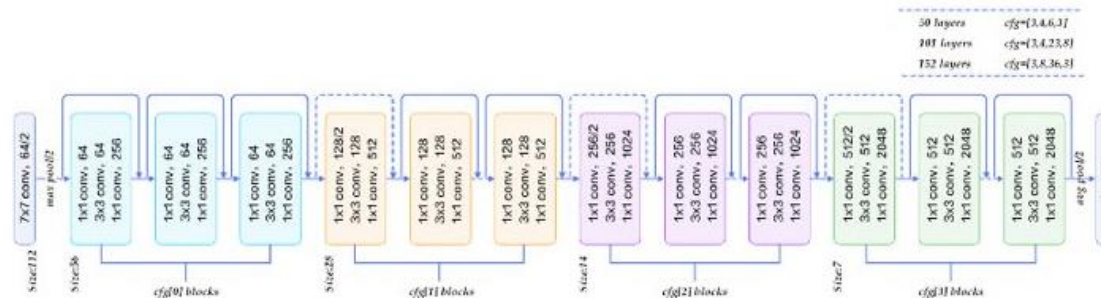
The winner of the ILSVRC 2014 competition was GoogleNet(a.k.a. Inception V1) from Google. It achieved a top-5 error rate of 6.67%! This was very close to human level performance which the organisers of the challenge were now forced to evaluate. As it turns out, this was actually rather hard to do and required some human training in order to beat GoogLeNets accuracy. After a few days of training, the human expert (Andrej Karpathy) was able to achieve a top-5 error rate of 5.1%(single model) and 3.6%(ensemble). The network used a CNN inspired by LeNet but implemented a novel element which is dubbed an inception module. It used batch normalization, image distortions and RMSprop. This module is based on several very small convolutions in order to drastically reduce the number of parameters. Their architecture consisted of a 22 layer deep CNN but reduced the number of parameters from 60 million (AlexNet) to 4 million.



2.6.3. Redes Neurais Convolucionais

ResNet(2015)

At last, at the ILSVRC 2015, the so-called Residual Neural Network (ResNet) by Kaiming He et al introduced anovel architecture with “skip connections” and features heavy batch normalization. Such skip connections are also known as gated units or gated recurrent units and have a strong similarity to recent successful elements applied in RNNs. Thanks to this technique they were able to train a NN with 152 layers while still having lower complexity than VGGNet. It achieves a top-5 error rate of 3.57% which beats human-level performance on this dataset.



AlexNet has parallel two CNN line trained on two GPUs with cross-connections, GoogleNet has inception modules ,ResNet has residual connections.

2.6.3. Redes Neurais Convolucionais

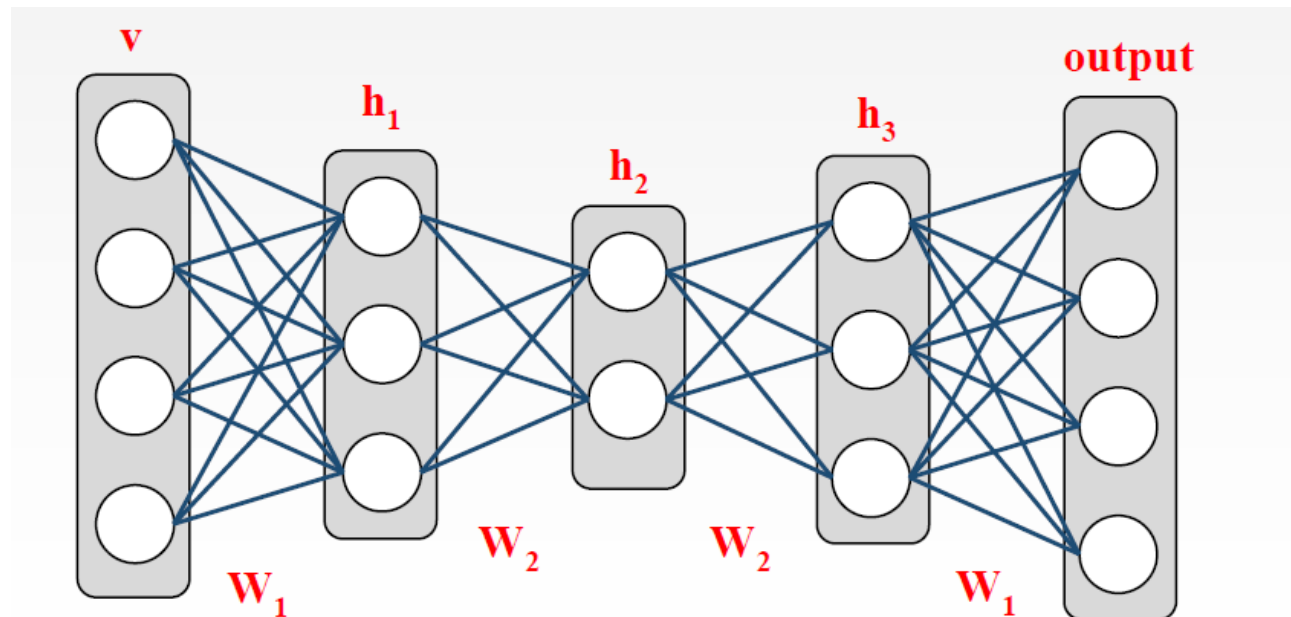
Summary Table

Year	CNN	Developed by	Place	Top-5 error rate	No. of parameters
1998	LeNet(8)	Yann LeCun et al			60 thousand
2012	AlexNet(7)	Alex Krizhevsky, Geoffrey Hinton, Ilya Sutskever	1st	15.3%	60 million
2013	ZFNet()	Matthew Zeiler and Rob Fergus	1st	14.8%	
2014	GoogLeNet(19)	Google	1st	6.67%	4 million
2014	VGG Net(16)	Simonyan, Zisserman	2nd	7.3%	138 million
2015	ResNet(152)	Kaiming He	1st	3.6%	

2.6.3. Redes Neurais Convolucionais

- **Autoencoder**

- Utilizado para aprender uma representação (encoding) para um conjunto de dados
 - » Geralmente empregado para redução de dimensionalidade
 - » Aprendizado não-supervisionado



2.6.3. Redes Neurais Convolucionais

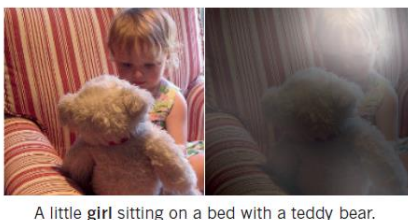
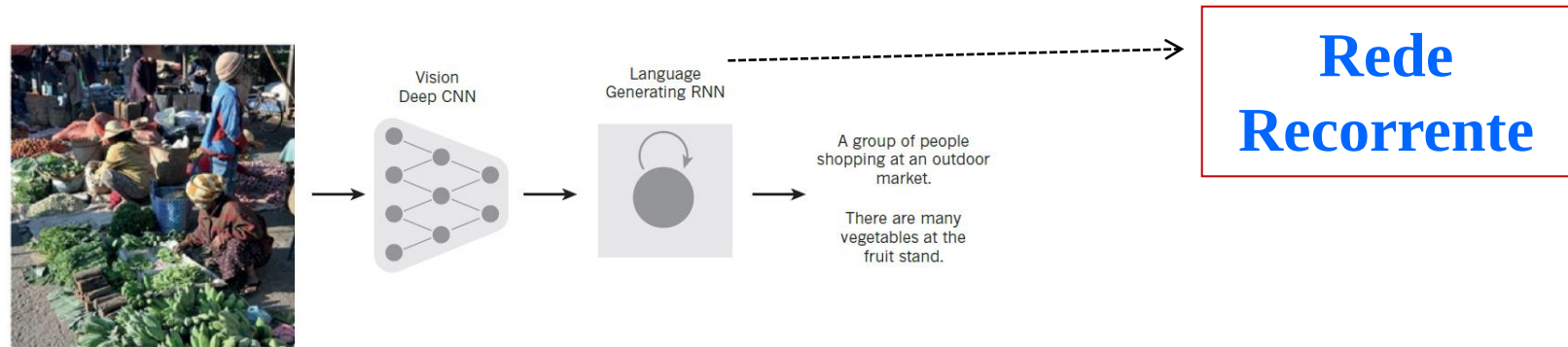


Figure 3 | From image to text. Captions generated by a recurrent neural network (RNN) taking, as extra input, the representation extracted by a deep convolution neural network (CNN) from a test image, with the RNN trained to 'translate' high-level representations of images into captions (top). Reproduced

with permission from ref. 102. When the RNN is given the ability to focus its attention on a different location in the input image (middle and bottom; the lighter patches were given more attention) as it generates each word (bold), we found⁸⁶ that it exploits this to achieve better 'translation' of images into captions.

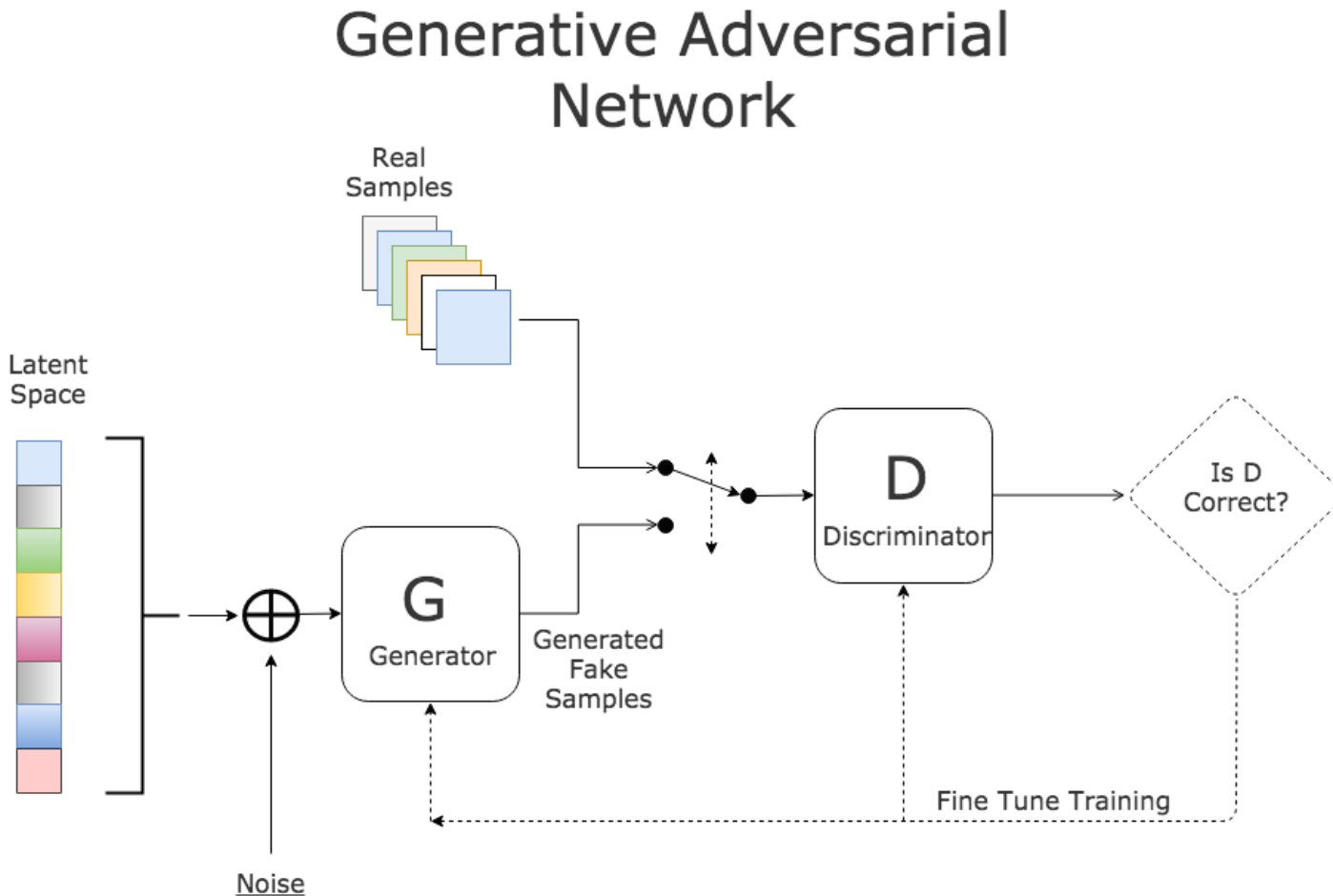
Fonte: LeCun, Y., Bengio, Y., & Hinton, G. (2015). "Deep learning", Nature, 521(7553), 436-444.

2.6.3. Redes Neurais Convolucionais



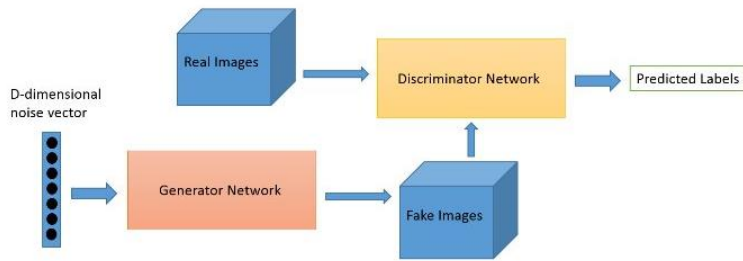
<https://futurism.com/incredibly-realistic-faces-generated-neural-network>

2.6.3. Redes Neurais Convolucionais



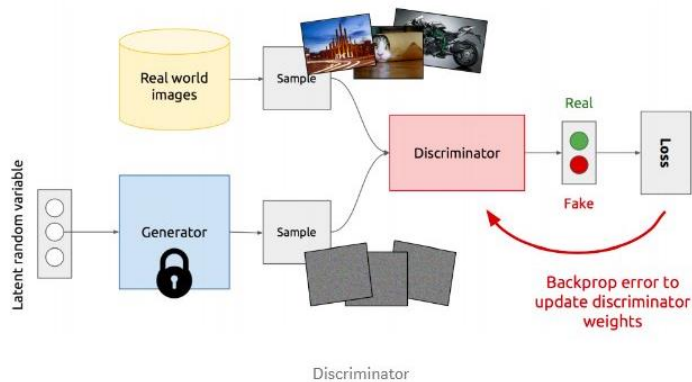
Fonte: <https://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html>

2.6.3. Redes Neurais Convolucionais

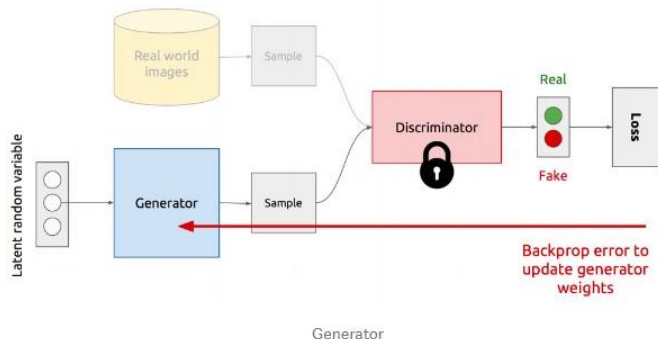


A simple flowchart of a GAN

Training Discriminator



Training Generator



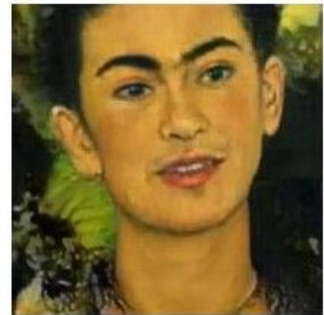
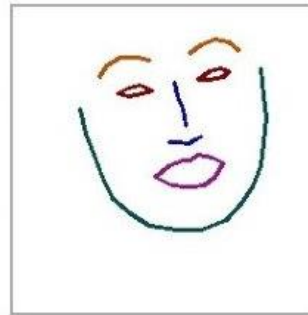
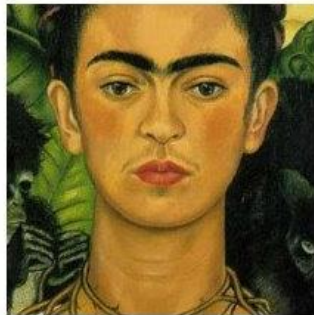
<https://youtu.be/6v7lJHFZZ4>

<https://towardsdatascience.com/art-of-generative-adversarial-networks-gan-62e96a21bc35>

2.6.3. Redes Neurais Convolucionais

Mona Lisa frown: Machine learning brings old paintings and photos to life

Devin Coldewey @techcrunch / 7:41 pm -03 • May 22, 2019



<https://techcrunch.com/2019/05/22/mona-lisa-frown-machine-learning-brings-old-paintings-and-photos-to-life/>

2.6.3. Redes Neurais Convolucionais

- Algumas Soluções

<http://caffe.berkeleyvision.org/>

Caffe

Deep learning framework
by [BAIR](#)

Created by
[Yangqing Jia](#)
Lead Developer
[Evan Shelhamer](#)

 [View On GitHub](#)

Caffe

Caffe is a deep learning framework made with expression, speed, and modularity in mind. It is developed by Berkeley AI Research ([BAIR](#)) and by community contributors. [Yangqing Jia](#) created the project during his PhD at UC Berkeley. Caffe is released under the [BSD 2-Clause license](#).

Check out our web image classification [demo!](#)

Why Caffe?

Expressive architecture encourages application and innovation. Models and optimization are defined by configuration without hard-coding. Switch between CPU and GPU by setting a single flag to train on a GPU machine then deploy to commodity clusters or mobile devices.

Extensible code fosters active development. In Caffe's first year, it has been forked by over 1,000 developers and had many significant changes contributed back. Thanks to these contributors the framework tracks the state-of-the-art in both code and models.

Speed makes Caffe perfect for research experiments and industry deployment. Caffe can process **over 60M images per day** with a single NVIDIA K40 GPU*. That's 1 ms/image for inference and 4 ms/image for learning and more recent library versions and hardware are faster still. We believe that Caffe is among the fastest convnet implementations available.

Community: Caffe already powers academic research projects, startup prototypes, and even large-scale industrial applications in vision, speech, and multimedia. Join our community of brewers on the [caffe-users group](#) and [Github](#).

* With the ILSVRC2012-winning [SuperVision](#) model and prefetching IO.

2.6.3. Redes Neurais Convolucionais

- **Algumas Soluções**

<https://www.tensorflow.org/?hl=pt-br>



Rápido

O desempenho é essencial ao criar e implantar sistemas de aprendizado de máquina. É por isso que o TensorFlow inclui o XLA, um poderoso compilador de álgebra linear que agiliza a execução do código do TensorFlow em processadores, CPUs, GPUs, TPUs e outras plataformas de hardware incorporadas.

Flexível

O TensorFlow fornece APIs de alto nível que facilitam a criação e o treinamento dos seus modelos, bem como o controle de baixo nível para máxima flexibilidade e desempenho.

Pronto para a produção

O TensorFlow pode ser usado em pesquisas exploratórias e até em produção em larga escala. É possível usar as mesmas APIs do TensorFlow em várias áreas, seja na criação de um novo tipo de modelo ou no processamento de milhões de solicitações na produção.

2.6.3. Redes Neurais Convolucionais

- Algumas Soluções

<https://keras.io/>

Keras: The Python Deep Learning library



Keras

You have just found Keras.

Keras is a high-level neural networks API, written in Python and capable of running on top of **TensorFlow**, **CNTK**, or **Theano**. It was developed with a focus on enabling fast experimentation. *Being able to go from idea to result with the least possible delay is key to doing good research.*

Use Keras if you need a deep learning library that:

- Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility).
- Supports both convolutional networks and recurrent networks, as well as combinations of the two.
- Runs seamlessly on CPU and GPU.

2.6. Deep Learning

- **Vídeos**

1. ***You Can't Play 20 Questions with Nature and Win***

https://www.youtube.com/watch?v=WXtYR2JEyEg&list=PLLP3KAwfEjgmF63Rg_mxLwevh1CrsFZkm&index=22

2. ***Large-Scale Deep Learning for Building Intelligent Computer Systems***

<https://www.youtube.com/watch?v=4hqb3tdk01k>

3. ***What's Wrong with Deep Learning?***

<http://techtalks.tv/talks/whats-wrong-with-deep-learning/61639/>

4. ***Does the Brain do Inverse Graphics?***

<https://www.youtube.com/watch?v=TFIMqt0yT2I>

5. ***Deep Learning, Self-Taught Learning and Unsupervised Feature Learning***

<https://www.youtube.com/watch?v=n1ViNeWhC24>

6. ***Visualizing and Understanding Deep Neural Networks***

<https://www.youtube.com/watch?v=ghEmQSxT6tw>

7. ***Ian Goodfellow: Generative Adversarial Networks (NIPS 2016 tutorial)***

8. https://www.youtube.com/watch?time_continue=9&v=HGYYEUSm-0Q&feature=emb_title

- Referências

- LeCun, Y., Bengio, Y., & Hinton, G. (2015). “Deep learning”, *Nature* 521(7553), 436-444.
- Best Resources for Getting Started With GANs

<https://machinelearningmastery.com/resources-for-getting-started-with-generative-adversarial-networks/>